

REMARKS

Reconsideration and allowance of the above identified patent application are hereby requested. Claims 1-9, 11-29, 31-33, and 35-38 are now in the application with claims 1, 11, 21, 31, and 32 being independent. The Applicants respectfully traverse the Office's rejections.

Rejection Under 35 U.S.C. §103(a)

Claims 1-9, 11-29, 31-33 and 36-38 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent No. 6,275,225 to Rangarajan et al. in view of U.S. Patent No. 5,907,326 to Atkin et al. Additionally, claim 35 stands rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over Rangarajan et al. in view of Atkin et al. and further in view of Sanna et al. Using Windows NT Workstation 4.0, 1997 ("Sanna"). These contentions are respectfully traversed.

CLAIM 1

The Office (Action of November 20, 2006 at p. 10) asserts (underlining in original) "...it is noted the features upon which applicant relies (i.e., a function description separate from an appearance description) are not recited in the rejected claim(s)." The Applicants respectfully disagree with this assertion. Claim 1 clearly indicates that a function description is separate from an appearance description.

Claim 1 recites (underlining added for emphasis) "...reading a function description of a first function to be provided by the user interface, the function description including logic for selecting an appearance of the user interface based on a geographic location of a user of the computer program on the fly at run time, the function description comprising instructions for handling user interface events; executing the logic included in the function description to select an appearance description of a first appearance to be presented by the user interface...." Thus, an appearance description is selected in accordance with logic included in the function description, and the selecting occurs on the fly at run time. Therefore, the function description is separate from the appearance description prior to the selection.

Further, claim 1 recites (underlining added for emphasis) "...associating the function description and the appearance description on the fly at run time into an executable form...."

Thus, claim 1 recites that the function description and the appearance description also are not associated until run time. Therefore, claim 1 clearly indicates that the function description is separate from the appearance description.

The Office (Action of November 20, 2006 at p. 3) asserts (underlining added for emphasis) "By the mere fact that the functionality and the appearance of the GUIs are both established at the same time does not imply that there are no separate function and appearance descriptions. The functionality and the appearance of the GUIs are both established at the same time when the configuration is saved. However, the modify procedure does allow separate modification of the functionality and the appearance of the GUIs, see col. 9, lines 41-47)."

Contrary to the Office's assertion, however, Rangarajan et al. fail to disclose, teach, or suggest a function description separate from an appearance description. Rangarajan et al. also do not teach separate modification of the functionality and the appearance of a GUI. Moreover, allowing separate modification of the functionality and appearance of a GUI is not equivalent to a function description separate from an appearance description.

Rangarajan et al. (Col. 9, lines 41-47) disclose (underlining added for emphasis) "A 'modify configuration' SCA 821 invokes the wizard process 300, previously described, but uses a reconstructed data structure that corresponds to the currently selected configuration (instead of the original data structure used when creating a new configuration) and selects the appropriate end node as the wizard's starting point." Thus, Rangarajan et al. teach that a configuration is represented using a single data structure. Rangarajan et al. also teach that the 'modify configuration' SCA uses the single reconstructed data structure that corresponds to the currently selected configuration. Therefore, Rangarajan et al. disclose a single data structure representing the function and appearance of a GUI, and not a separate function description and appearance description.

Further, Rangarajan et al. do not disclose allowing separate modification of the functionality and the appearance of the GUIs. To the contrary, Rangarajan et al. (*Id.*) disclose using the wizard process 300 to modify the configuration. Rangarajan et al. (Col. 6, lines 7-11) state (underlining added for emphasis) "...the wizard process 300 continues to a 'create configuration' decision procedure 304 that determines whether the invocation of the wizard process 300 was for the purpose of creating a new configuration or modifying an existing

configuration." Thus, the same wizard process 300 is used both to create new configurations and to modify existing configurations. Rangarajan et al. (Col. 6, lines 18-22) also disclose (underlining added for emphasis) "...a 'select configuration end node' procedure 306 locates the end node in the wizard data structure that matches the configuration to be modified and reconstructs the state of the wizard data structure to match the actual configuration." As such, the single data structure that represents the configuration, including the functionality and the appearance is accessed during modification.

Additionally, Rangarajan et al. (Col. 6, lines 22-27) disclose (underlining added for emphasis) "Once a node is selected, a 'present node interrogatory' procedure 307 presents the node's interrogatory to the user. This interrogatory may include a representation of the problem domain to allow the user to select which aspects of the problem domain the user desires to access from the GUI." Thus, the interrogatory allows the user to specify the functionality. Rangarajan et al. (Col. 6, lines 54-59) further disclose (underlining added for emphasis) "...if the 'end node' decision procedure 313 determines that the current nodes [sic] is an end node, the wizard process 300 continues to a 'construct GUI configuration' procedure 317 that constructs the customized GUI configuration dependent on the user's responses to the interrogatories." Thus, Rangarajan et al. teach that for both creating a new configuration and for modifying an existing configuration the appearance (GUI) is constructed dependent on the user's responses to the interrogatories (functionality). Therefore, the modify procedure does not allow separate modification of the functionality and the appearance of the GUIs, as the Office asserts.

Moreover, Rangarajan et al. and Atkin et al. fail to disclose associating the function description and the appearance description on the fly at run time into an executable form, as is claimed.

The Office (Action of November 20, 2006 at p. 3) asserts that Rangarajan et al. teach "...associating the function description and the appearance description on the fly at run time into an executable form; and executing the executable form of the user interface to generate the user interface with the associated function description and appearance description (col. 5, lines 32-46; 215 of fig. 2; col. 7, line 62 to col. 8, line 12; and col. 9, lines 49-52;...." However, Rangarajan et al. fail to disclose all of the claimed subject matter.

Rangarajan et al. (Col. 5, lines 32-46) disclose that a configuration can be saved as the default configuration after it is generated and that the initialization process 200 can load the default configuration. Rangarajan et al. (*Id.*) also disclose (underlining added for emphasis) "After loading the configuration, the initialization process 200 continues to a 'create GUI according to selected configuration' procedure 215 that processes the information stored in the selected configuration to create a user configured GUI that contains the selected feature set." Thus, the configuration is saved as a single entity prior to the initialization process that creates the GUI. Therefore, Rangarajan et al. do not disclose, teach, or suggest associating the function description and the appearance description on the fly at run time into an executable form.

Further, Rangarajan et al. (Col. 7, line 62 to Col. 8, line 12) disclose that the configuration specified by the user's answers to interrogatories is saved in accordance with a user specified name and description. Rangarajan et al. (*Id.*) also disclose "Once the user selects a finish SCA 611, the currently defined configuration is saved." Additionally, Rangarajan et al. (Col. 9, lines 49-52) disclose (underlining added for emphasis) "A 'select and run' SCA 825 makes the selected configuration the current configuration and invokes the console display using the selected configuration while preserving the default configuration." Rangarajan et al. (Abstract) further disclose (underlining added for emphasis) "The invention engages in a dialog with the user to determine which GUI features are desired by the user. Depending on the user's answers a model representation of the GUI (a configuration) is saved." Thus, Rangarajan et al. teach that the single, selected configuration is used to generate the display. Therefore, Rangarajan et al. fail to disclose, teach, or suggest associating the function description and the appearance description on the fly at run time into an executable form, as is claimed.

Additionally, Atkin et al. teach changing an appearance during operation. For example, Atkin et al. (Col. 6, lines 20-26) disclose (underlining added for emphasis) "In the present invention, the locale change daemon takes the form of a drag and drop presentation manager application. The locale change daemon stays suspended until a user clicks on it with a mouse operation. When the daemon is invoked, a locale change may take place by simply dragging a locale change object and dropping it on top of an application." Thus, Atkin et al. teach that the user can invoke the locale change daemon only after the application has begun executing, not at

run time. Therefore, Atkin et al. also do not disclose, teach, or suggest associating the function description and the appearance description on the fly at run time into an executable form.

For at least these reasons, independent claim 1 is allowable over Rangarajan et al. in view of Atkin et al. Claims 2-9, 37, and 38 depend from claim 1 and are allowable for at least the reasons discussed with respect to claim 1.

CLAIM 2

Claim 2 recites (underlining added for emphasis) "...further comprising replacing the function description during program execution while providing a continuity of presentation."

The Office (Action of November 20, 2006 at p. 4) asserts that "Rangarajan teaches replacing the function description during program execution while providing continuity of presentation (col. 8, lines 57-65...." The Office (*Id.*) also cites to a description of the profile-manager display at Col. 9, lines 12-37. Rangarajan et al. fail to disclose all of the claimed subject matter.

Rangarajan et al. (Col. 8, lines 57-63) disclose (underlining added for emphasis) "If the modify entry SCA is activated, a modify procedure 719 is invoked that recreates the wizard data structure that resulted in the selected configuration. Next the wizard is invoked indicating the terminal node in the data structure that generated the configuration. Thus the user is able to back track over the previous interrogatories and modify them as appropriate." Thus, Rangarajan et al. teach that the modify procedure 719 invokes the wizard process for modifying an existing configuration.

With respect to the wizard process, Rangarajan et al. (Col. 6, 54-59) disclose (underlining added for emphasis) "...if the 'end node' decision procedure 313 determines that the current nodes [sic] is an end node, the wizard process 300 continues to a 'construct GUI configuration' procedure 317 that constructs the customized GUI configuration dependent on the user's responses to the interrogatories." As such, Rangarajan et al. teach that a customized GUI is constructed dependent on the user's responses to the interrogatories. Thus, changed responses to one or more interrogatories are reflected in the corresponding customized GUI. Therefore, Rangarajan et al. do not disclose replacing the function description during program execution while providing a continuity of presentation.

Further, Rangarajan et al. (Col. 9, lines 12-37) disclose a profile-manager display for managing stored configurations. Rangarajan et al. (*Id.*) also disclose that the profile-manager display "...includes a selection of SCAs that allow the user to invoke operations." Invoking an operation through an SCA is not equivalent to replacing the function description. For example, Rangarajan et al. (Col 1, lines 19-27) disclose (underlining added for emphasis) "The user moves a cursor, responsive to a pointing device, over a selectable control area (SCA) on the display....When the computer application supports a large set of features, the number of SCAs, associated with the features, becomes unwieldy." Thus, Rangarajan et al. teach that SCAs represent features accessible through the GUI. Rangarajan et al. do not, however, disclose, teach, or suggest replacing the function description during program execution while providing a continuity of presentation, as is claimed.

For at least these reasons, dependent claim 2 also is allowable over Rangarajan et al. in view of Atkin et al. based on its own merits.

CLAIM 11

The Office (Action of November 20, 2006 at p. 12) also asserts that claim 11 does not recite a map component separate from a fashion component (it is noted that the Office repeats the terms "function description" and "appearance description" used in claim 1). The Applicants respectfully disagree with this assertion. Claim 11 clearly indicates a map component separate from a fashion component.

Claim 11 recites (underlining added for emphasis) "selecting a map component and a fashion component, wherein at least one of the map component and the fashion component are selected automatically according to a geographic location of a user of the computer program on the fly at run time; associating the map component and the fashion component to generate the user interface, the map component including logic for changing one of the map component and the fashion component, the map component including instructions for handling and processing interface objects and events; combining the map component and the fashion component into an executable form; and executing the user interface with the associated map component and fashion component."

Thus, claim 11 recites selecting both a map component and a fashion component; associating the map component and the fashion component; and combining the map component and the fashion component. Interpreting claim 11 such that the map component and the fashion component are not separate improperly renders the associating and combining the map component and the fashion component superfluous. Thus, claim 11 clearly recites a map component separate from a fashion component. As discussed above with respect to claim 1, Rangarajan et al. and Atkin et al. fail to disclose a map component separate from a fashion component.

Further, claim 11 recites (underlining added for emphasis) "...associating the map component and the fashion component to generate the user interface, the map component including logic for changing one of the map component and the fashion component, the map component including instructions for handling and processing interface objects and events...."

The Office (Action of November 20, 2006 at pp. 4-5) asserts "Rangarajan teaches a method of defining a user interface for a computer program, comprising:...associating the map component and the fashion component to generate the user interface (fig. 8; col. 9, lines 22-34 and col. 9, lines 49-52), the map component including logic for changing one of the map component and the fashion component (col. 9, line 59 to col. 10, line 13)...." Rangarajan et al., however, fail to disclose all of the claimed subject matter.

Rangarajan et al. (Col. 9, line 59 – Col. 10, line 13) disclose that a GUI contains a topology hierarchy pane, a topology view pane, and an alarm summary pane. Rangarajan et al. (*Id.*) further disclose that a "device detail display" and an "alarm summary pane" are available through the GUI. Additionally, Rangarajan et al. (*Id.*) disclose (underlining added for emphasis) "...the information required to generate the above displays is obtained (directly or indirectly) from networked devices." Generating a display is not equivalent to changing one of the map component and the fashion component, as is claimed.

Rangarajan et al. (Col. 9, lines 41-43) disclose "A 'modify configuration' SCA 821 invokes the wizard process 300...." As discussed above, Rangarajan et al. (Col. 6, lines 7-11) disclose (underlining added for emphasis) "...the wizard process 300 continues to a 'create configuration' decision procedure 304 that determines whether the invocation of the wizard process 300 was for the purpose of creating a new configuration or modifying an existing

configuration." Thus, Rangarajan et al. teach that a configuration (i.e., function and appearance) is changed using the wizard process 300. Further, Rangarajan et al. (Col. 6, lines 18-20) disclose (underlining added for emphasis) "...a 'select configuration end node' procedure 306 locates the end node in the wizard data structure that matches the configuration to be modified...." Thus, Rangarajan et al. teach that the wizard (i.e., the logic) used to modify the configuration is separate from the configuration. Therefore, Rangarajan et al. do not disclose, teach, or suggest the map component including logic for changing one of the map component and the fashion component, as is claimed.

Additionally, claim 11 recites (underlining added for emphasis) "...selecting a map component and a fashion component, wherein at least one of the map component and the fashion component are selected automatically according to a geographic location of a user of the computer program on the fly at run time...."

The Office (Action of November 20, 2006 at p. 5) concedes that "Rangarajan does not explicitly teach [] selecting at least one of a map component and a fashion component according to a geographic location of a user of the computer program on the fly at run time." The Office (*Id.*), however, asserts "Atkin teaches [] selecting at least one of a map component and a fashion component according to a geographic location of a user of the computer program on the fly at run time (see e.g., abstract; col. 3, line 64 to col. 4, line 4; and col. 4, lines 19-26)."

Further, the Office does not assert that Atkin et al. teach that the selecting is performed automatically. Instead, the Office (*Id.* at p. 12) asserts "...broadly providing an automatic or mechanical means to replace a manual activity which accomplished the same result is not sufficient to distinguish over the prior art." However, in addition to not being performed automatically, the system and method disclosed by Atkin et al. also do not accomplish the same result as the claimed subject matter.

Atkin et al. do not disclose, teach, or suggest that at least one of the map component and the fashion component are selected according to a geographic location of a user of the computer program. To the contrary, Atkin et al. (e.g., Abstract) disclose that a user can select one or more cultural profiles in accordance with the user's preference.

For example, Atkin et al. (Col. 4, lines 19-26) disclose (underlining added for emphasis) "The present invention allows programs to be customized to support any culture or combination

of cultures. This is accomplished through the use of a drag and drop interface,...where cultural profiles...are dropped onto windows...and the window then updates its cultural profile information accordingly." Thus, Atkin et al. teach that the cultural profile of a window is not selected based on a geographic location of a user.

Further, Atkin et al. (Col. 6, lines 42-50) disclose (underlining added for emphasis) "the system and method of the present invention allow different windows, within a multi-window application, to use different cultural profiles...the user can work with a spreadsheet that has an Arabic interaction cultural profile and a Japanese data cultural profile. In addition, the user may wish help windows within the spreadsheet application to use French language...with a Mexican toolbar...." Thus, Atkin et al. teach that the user manually specifies one or more cultural profiles based on user preference, not geographic location. Therefore, Atkin et al. and Rangarajan et al. do not disclose, teach, or suggest selecting a map component and a fashion component, wherein at least one of the map component and the fashion component are selected automatically according to a geographic location of a user of the computer program on the fly at run time.

For at least these reasons, independent claim 11 is allowable over Rangarajan et al. in view of Atkin et al. Claims 12-20 depend from claim 11 and are allowable for at least the reasons discussed with respect to claim 11.

CLAIM 21

Claim 21 is similar to claim 1. For example, claim 21 recites (underlining added for emphasis) "...read a function description of a first function to be provided by the user interface, the function description including logic for selecting an appearance of the user interface based on a geographic location of a user of the computer program on the fly at run time; execute the logic included in the function description to select an appearance description of a first appearance to be presented by the user interface; associate the function description and the appearance description on the fly at run time; and execute the user interface with the associated function and appearance." Therefore, claim 21 is allowable for at least the reasons discussed with respect to claim 1. Additionally, claims 22-29 depend from claim 21. Therefore, claims 22-29 are at least allowable based on claim 21.

CLAIM 31

Claim 31 is similar to claim 11. For example, claim 31 recites (underlining added for emphasis) "...select a map component and a fashion component, wherein at least one of the map component and the fashion component are selected automatically according to a geographic location of a user of the computer program on the fly at run time; associate the map component and the fashion component to generate the user interface, the map component including logic for changing one of the map component and the fashion component; and execute the user interface with the associated map component and fashion component." Therefore, claim 31 is allowable for at least the reasons discussed with respect to claim 11.

CLAIM 32

Claim 32 is similar to claim 11. For example, claim 32 recites (underlining added for emphasis) "...means for selecting a map component and a fashion component, wherein at least one of the map component and the fashion component are selected automatically according to a geographical locale of a user of the computer program on the fly at run time; means for associating the map component and the fashion component on the fly at run time to generate the user interface, the map component including logic for changing one of the map component and the fashion component; means for executing the user interface with the associated map component and fashion component; and means for changing the fashion component to present logic of the user interface with a different appearance during program execution."

Therefore, claim 32 is allowable for at least the reasons discussed with respect to claim 11. Additionally, claims 33, 35, and 36 depend from claim 32 and are at least allowable based on claim 32.

CLAIM 35

Claim 32 recites "...a device coupled to the processor to present the user interface...." Further, claim 35 recites "wherein the device is a telephone." Thus, claim 35 is directed to a telephone (the device) coupled to the processor to present the user interface.

The specification (page 4, lines 14-16) further clarifies that (underlining added for emphasis) "The appearances can be presented using a graphical interface, or can be presented in a non visual modality such as a voice user interface that can be used in speech driven computers, including telephones and voice activated computers."

The Office (Action of November 20, 2006 at 14) asserts (underlining added for emphasis) "...claim 35 calls for a device (or telephone) coupled to the processor to process the user interface, however, claim 35 does not call 'for the device – the telephone – to present the user interface' as asserted by applicant." The Office's interpretation is not supported by the language of claims 32 and 35, and is respectfully traversed.

Claim 35 does not call for a telephone (the device) coupled to the processor to process the user interface, as the Office asserts. To the contrary, claim 35 calls for a telephone (the device) coupled to the processor to present the user interface.

Sanna (page 442) discloses (underlining added for emphasis) "Windows NT 4.0 includes a phone dialer, a new application you can use to dial your phone." Sanna (*Id.*) further discloses (underlining added for emphasis) "To use Phone Dialer, follow these steps: 1. Choose Programs from the Start menu. 2. Place your pointer on Accessories. 3. Choose Phone Dialer from the submenu. The Phone Dialer application launches." Thus, Sanna discloses that a computer application, the phone dialer, presents a user interface. Sanna does not, however, disclose that a phone presents a user interface. Therefore, Sanna does not disclose, teach, or suggest a telephone coupled to the processor to present the user interface, as is claimed.

For at least these reasons, dependent claim 35 also is allowable over Rangarajan et al. in view of Atkin et al. and further in view of Sanna based on its own merits.

Concluding Comments

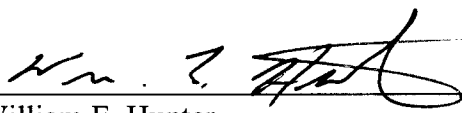
The specification (page 4, lines 3-8) discloses that "The logic and appearance of a user interface can be built independently, making the UI presentation more accessible to non-programmers. The UI can be modified without re-compiling the application. Because of late binding, the details of the UI presentation are established at execution time for the application. Hence, user interface development can be decoupled from code development. Hence, a visual designer can design a user interface without incurring delays associated with involving a programmer." In view of the remarks presented above, it should be clear that the art of record fails to teach or suggest the claimed subject matter so as to realize these potential advantages.

The foregoing comments made with respect to the positions taken by the Examiner are not to be construed as acquiescence with other positions of the Examiner that have not been explicitly contested. Accordingly, the above arguments for patentability of a claim should not be construed as implying that there are not other valid reasons for patentability of that claim or other claims.

No fees are believed to be due at this time. Please apply any charges not covered, or credits, to deposit account 06-1050.

Respectfully submitted,

Date: January 22, 2007



William E. Hunter
Reg. No. 47,671

Fish & Richardson P.C.
PTO Customer No. **21876**
Telephone: (858) 678-5070
Facsimile: (858) 678-5099